European Scientific Journal March edition vol. 8, No.6 ISSN: 1857 – 7881 (Print) e - ISSN 1857- 7431

QUALITATIVE SYNERGISM OF MULTI-AGENT SOFTWARE ARCHITECTURE AND SYSTEM REQUIREMENTS

Shahbaz Nazeer

Government College University Faisalabad, Pakistan

Tahir Abdullah

Government College University Faisalabad, Pakistan

Rabia Saleem

Government College University Faisalabad, Pakistan

Abstract

Software architecture design and requirement engineering are core and independent areas of engineering. A lot of research, education and practice are carried on Requirement elicitation and doing refine it, but it is a major issue of engineering. QSMSR (QUALITATIVE SYNERGISM OF MULTI-AGENT SOFTWARE ARCHITECTURE AND SYSTEM REQUIREMENTS) model act as a bridge between requirement and design there is a huge gap between these two areas of software architecture and requirement engineering. The principal model defines how to take input the requirements and to refine it in such a way that the gap is covered.The qualitative model merge fabricated model and classified model. Classified model make the sub groups of the role and match it with components. The Fabricated model link QSMSR Principal Model to an architecture design. These both model (principal model, qualitative model) are the sub models of QSMSR.

Keywords: QSMSR (Qualitative Synergism of Multi-agent Software Architecture and Requirement engineering), Software Architecture, Requirement Engineering.

Introduction

The relationship between architecture and requirements of a system to be is neither clear nor understandable, stakeholders may have contradictory goals nor expectations, non-functional requirements are tough to be mapped to an architectural entity, (Chung *et al.,* (2000)).

Software architecture requirements engineering are well-known fields of research, education and practice in the software engineering society. Because of the significant progress on these two fronts, we still need the solid basis, technique and tools to support the synergism achievement of architectural objectives within the context of complex stakeholder associations.

The basic concepts of security in computing, and some characteristics of agents and multi-agent systems that introduce new threats and ways to attack After this, some models and architectures proposed in the literature are presented and analyzed (Cavalcante. R, (2011))

These requirements are often vague, unfinished, incompatible, and usually expressed unceremoniously. By contrast, requirement activities focus on the totality, reliability and confirmation of the requirements. Early stage requirements engineering activities have objectives and suppositions that are different from those of the later stage. (Alencar*et al*, (2001))

QSMSR Model:

QSMSR (QUALITATIVE SYNERGISM OF MULTI-AGENT SOFTWARE ARCHITECTURE AND SYSTEM REQUIREMENTS) model act as a bridge between requirement and design there is a huge gap between these two areas of software architecture and requirement engineering. It further divides its functionality in two sub models principal model and qualitative model which are explain further.

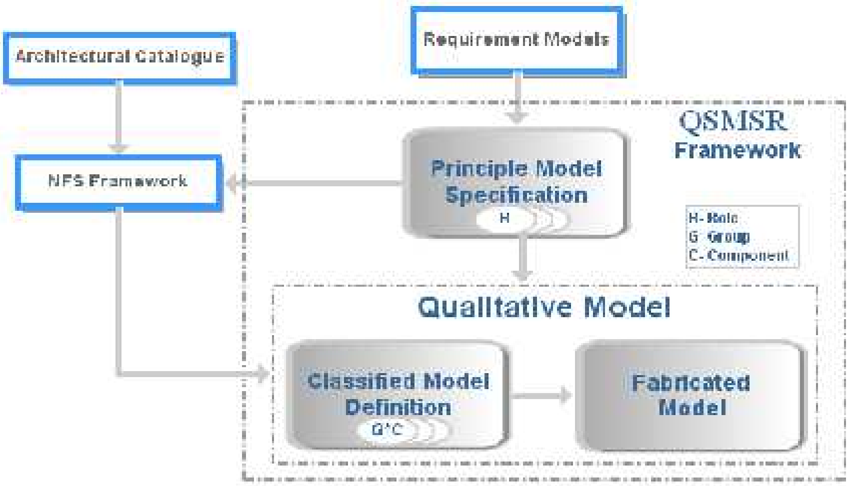

Figure 1: The QSMSR Framework (Detail Description)

1.1 Principal Model:

It defines the roles and iteration of the roles. After getting the requirement it provide the QSMSR principal model as the output of the system. In QSMSR we are focusing on the principal model. The principal model gets the requirements model as the input and than produce architectural catalog. This catalog is further use for much purpose but QSMSR use it for purposing the architectural design of the system. The principal divided into three sub task goal task refinement and role identification and then at the last selection of the architectural selection. These are three main tasks of the principal model of QSMSR model.

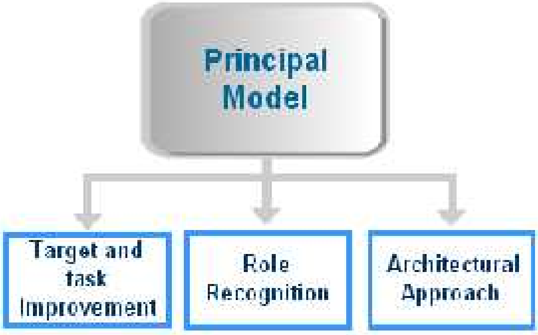

Figure 2: Principal Model

In the goal refinement firstly we analysis the actors and their role after that we refine these goals by their contribution of the system. In this we check which actor involved which type of role and how he interact with the system for this purpose we use the OR Decomposition, AND Decomposition and contribution.

As show in the below figure how a actor contribute to the system. It show how contribute it positively or negatively according to that we select the contribution and refine it. It is root of the sub system.

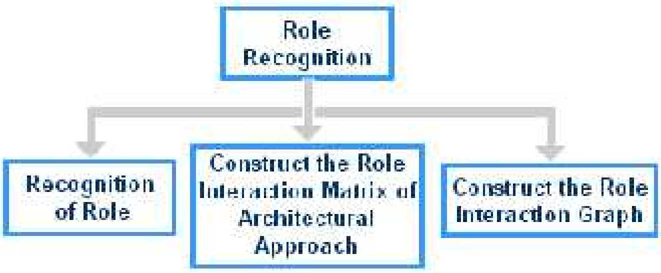

Figure 3: Role identification

In the role identification we define the role of the actors of the system. In MAS the scenario is totally different its too much complex to having the role identification. For this there are some specific tasks are followed by the role identification. Tasks are group together and show depends of these to each other's and similar task are group into one and different task are group in different group. In this we define the roles and define the relation between the roles how roles are interact to each other .in this we check the low level of coupling of the roles. The groups are refining her again as per role and iteration to each other and these are the selected goals that are we accomplish further. The process is processed as further.

The architectural selection has two sub task centrality equivalence and similarity equivalence.

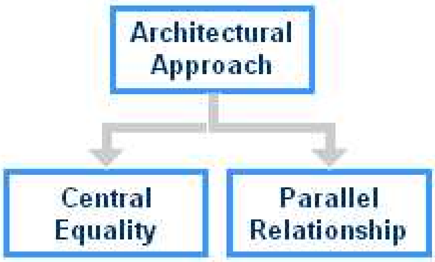

Figure 4: Architectural selection

Architectural selection depends on two task centrality equivalence and the similarity equivalence. In first task we find out the centurial actor of the given problem. For this purpose we draw the role interaction graph and for that graph we calculate the in degree and out degrees of our actor and the actor mostly have the in and out degree we consider it as the as our centurial actor. In second we check is there any similar actor existed which doing the same task in the system if exist than we eliminate that actor.

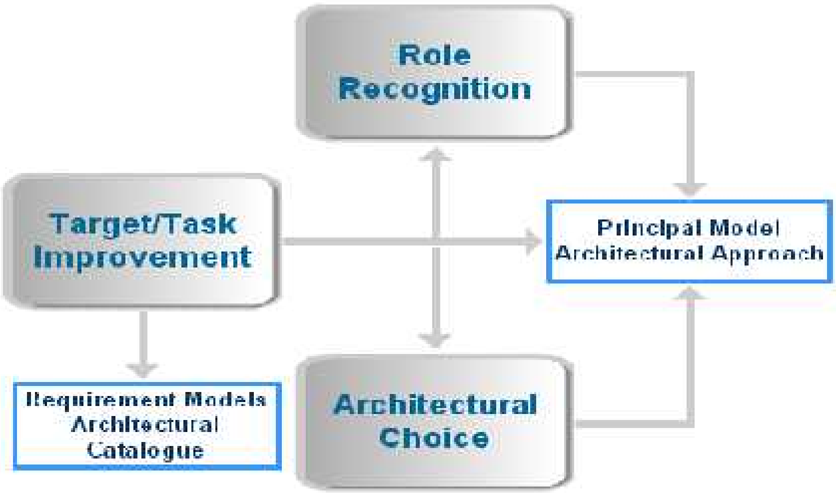

Figure 5: The Principal Model Specification

The process of the QSMSR principal model is defined in above figure in principal model firstly we get the goal and task refinement from the requirement model. For this the QSMSR principal model define the clustering and in other sense you can say define the sub task of the system and than refine it. After the refining process .The principal model gets it as the input for this model. than model applied correlation and clustered analysis on it for this we use the perason correlation formula we apply this formula to all sub task and take the correlation of that sub task and in the same time we have some architectural design or architectural style where the our correlation results are matched. Than we calculate the correlation of the architectural style and match this correlation to our problem than we convert it in to the architectural style. Suppose we have the example where we have the 7 actors but our correlation matched with structure-in-5.and in it we have the 5 location. Than we analysis the problem and check which actors are doing more likely to same work than we merge these actor .for example we have two actor journal reviewer and review handler for the example of journal publisher by seeing these actors as their abstract view we combine it into one and give the name review controller that further have two sub actors handler and reviewer.

1.2 Qualitative Model:

It takes the QSMSR principal model and generates the architectural model for the defined system. To complete that the model is further divided into two sub systems as follow.

In the Qualitative model, roles are clustered into sub-groups. These sub-groups are related to components, based on their similarity with the architectural components. The result is a Classified model, which is the allocation of sub-groups to architectural components.

Our framework advocates that a system corresponds to an organizational structure, in which actors are members of a group, playing roles in order to perform specific tasks. Roles can be used both as an intuitive concept in order to analyses requirements in multi-agent systems as well as a behavioral structure in order to derive coherent software architectures.

Furthermore, the QSMSR Framework provides the QSMSR Process to accomplish the transition from classified model from requirements.

The qualitative model is also further divided in two sub model are called fabricated model and classified model. Links the QSMSR Principal Model to an architectural style Assigns sub-groups to architectural components and generates the architectural model of the MAS. Takes Architectural Configuration as input and generates the Architectural Configuration of the MAS.

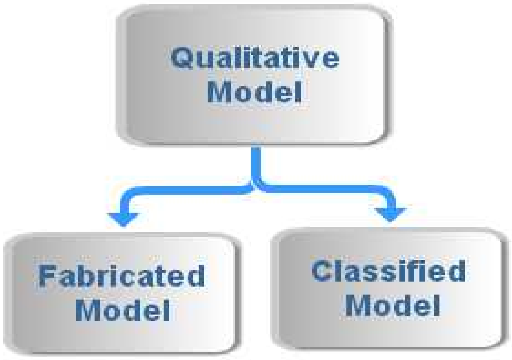

Figure 6: Qualitative model

The classified divide its activities further in two sub activities correlation analysis and cluster analysis. Clusters roles into sub-groups and matches sub-groups to components. Takes the Principal Model as input and generates the Architectural Configuration as an output.

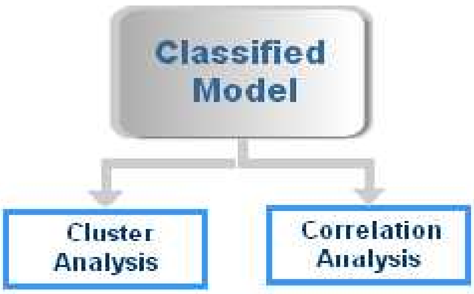

Figure 7: Classified models

The cluster analysis activities are Similarity Correlation Analysis, Centrality Correlation Analysis in this don the correlation and cluster analysis with help of formulas.

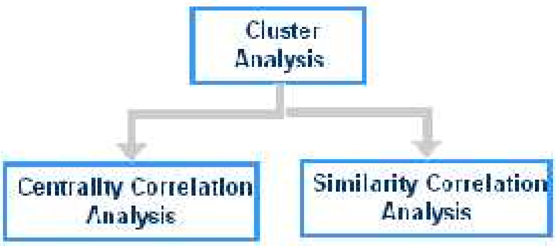

Figure 8: Cluster Analysis

The correlation analysis activities are Similarity equivalence, Centrality Correlation Analysis in this don the correlation and cluster analysis with help of formulas.

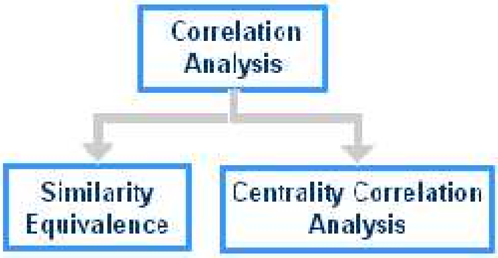

Figure 9: Correlation Analysis

1.2.1 Classified Model:

It converts the system in to sub group and match these subgroups to components of the system. It defines the sub system according to their roles of the system. It takes the QSMSR principal model as an input and generates the architectural configuration of the system.

1.2.2 Fabricated Model:

It takes the QSMSR Architectural Configuration as an input and allocates the architectural model .it compares the architectural sub configuration with architectural sub model and then the model selected which is best in the best.

The QSMSR Process:

In software development, to develop the architectural design from the traditional and detail requirement is too much complex. To do this we have to define some activities are analysis and to accomplish each decision involved.

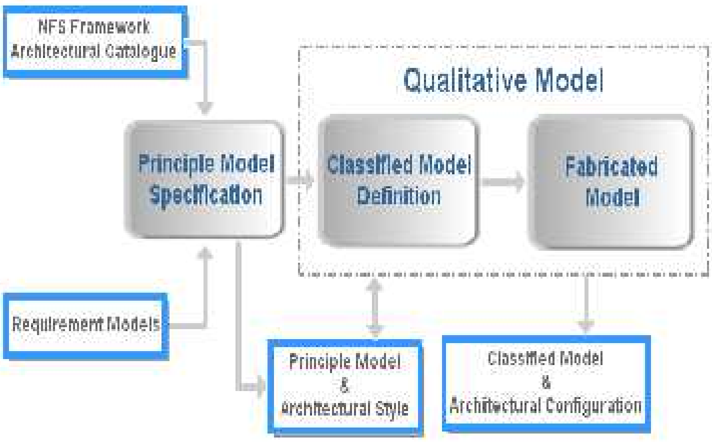

Figure 10: The QSMSR Process Activities

Process of Principal Model:

The first activity, principal Model takes the requirements and generates the architectural catalog of the system. To do this we follow some process as follow in the below figure. Our principal model is further divided in to role indemnification, goal refinement, architectural selection by doing this process we develop the architectural style.

Goal Refinement:

In the goal refinement firstly we analysis the actors and their role after that we refine these goals by their contribution of the system. In this we check which actor involved which type of role and how he interact with the system for this purpose we use the OR Decomposition, AND Decomposition and contribution.

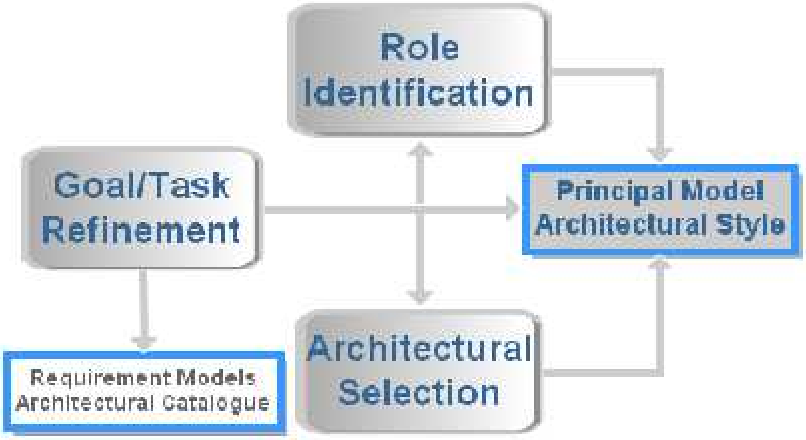

Figure 11: The Principal Model Specification

As show in the below figure how a actor contribute to the system. It show how contribute it positively or negatively according to that we select the contribution and refine it. It is root of the sub system

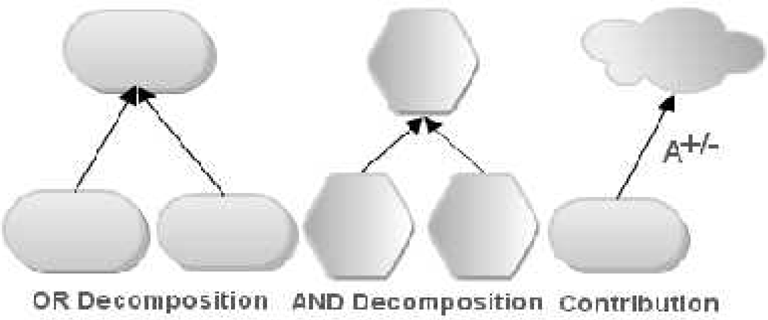

Figure 12: Tropos representation of goal refinement with means end and contribution analysis.

3.3.3 Role identification:

In the role identification we define the role of the actors of the system. In MAS the scenario is totally different its too much complex to having the role identification. For this there are some specific tasks are followed by the role identification.

3.4.2.1 Task cohesion:

Tasks are group together and show depends of these to each other’s and similar task are group into one and different task are group in different group.

3.4.2.2 Low role coupling:

In this we define the roles and define the relation between the roles how roles are interact to each other .in this we check the low level of coupling of the roles.

3.4.2.3 Goal coherence:

The groups are refining her again as per role and iteration to each other and these are the selected goals that are we accomplish further.

3.4.3 Architectural Selection:

Software Architected focus on the research and match the requirements architecture and which design have high value match is selected it is for the sub group of the system. We show its participation in the table as following.

A^{+} = PARTIAL/POSITIVE

A^{++} = PARTIAL/NEAGATIVE

A^{-} = SUFFICENT/POSITIVE

A^{--} = SUFFICENT /NEAGATIVE

Table 1: Architecturl Selection

Flat	A^{--}	A^{--}	A^{-}			A^{+}	A^{+}	A^{++}	A^{-}
Struct-5	A^{+}	A^{+}		A^{+}	A^{--}	A^{+}	A^{++}	A^{++}	A^{++}
Pyramid	A^{++}	A^{++}	A^{+}	A^{++}	A^{--}	A^{+}	A^{--}	A^{-}	
Joint-Vent	A^{+}	A^{+}	A^{++}	A^{+}	A^{-}	A^{++}		A^{+}	A^{++}
Bid	A^{--}	A^{--}	A^{++}	A^{-}	A^{++}	A^{-}	A^{--}	A^{++}	
Takeover	A^{++}	A^{++}	A^{-}	A^{++}	A^{--}	A^{+}		A^{+}	A^{+}

Arm's-Length	A^-	A^{--}	A^+	A^-	A^{++}	A^{--}	A^{++}	A^+	
Hierarchical			A^+	A^+	A^+	A^+		A^+	A^+
Vertical Integration	A^+	A^+	A^-	A^+	A^-	A^+	A^{--}	A^{--}	A^{--}
Cooptation	A^-	A^-	A^{++}	A^{++}	A^+	A^{--}	A^+		

According to this if our task group matches high nodes we called it sufficiently positive and show it as A^{++}. If our task group matches, medium nodes we called it partial positive and show it as A^+. If our task groups no matches, medium nodes we called it partial negative and show it as A^-. If our task groups no match's high nodes we called it sufficiently positive and show it as A^{--}.if we can do the matches on to our criteria we put in the blank space.

3.5 Process of Qualitative model:

The qualitative model further divided in to the sub models and these models provide the synergy between requirements and MAS. The two models are classified model and fabricated model. Which descriptions are further given in detail?

3.5.1 Process of Classified model:

The classified model makes the group of roles in to sub group. Which are than compared with the architectural design of the problem domain system?

3.5.1.1 Centrality:

It show the roughly estimation of the actor of the system. it basically show how much they are connected to each other on bases of these it show the value and the worth of actor in the system .it show how much a system participate in the system and how its works with the system.

3.5.1.2 Structural Equivalence:

In this we check how many actors having the same type of relation with other actors by this we can refine our process to identify there role of the system.

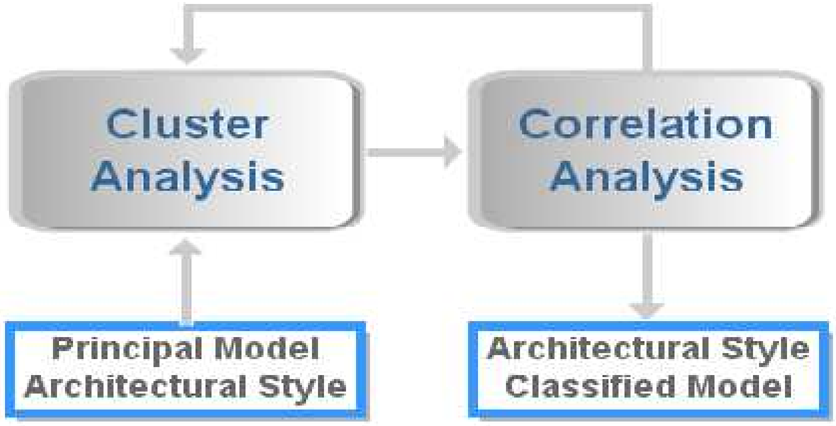

Figure 13: Classified model Definition

3.5.2 Cluster Analysis:

In this section we analysis. the cluster that are made is of similar roles It will check that same type of role of a cluster are how much match to each other in this we make the centurial role according to the define their iteration with other system .if it can be the central role we make it central otherwise we connect it to other cluster. By doing this we minimize the complexity of the system. It takes principal model as input provides the refine cluster.

3.5.3 Correlation Analysis:

This depends own the QSMSR sub group and architectural design. We analysis the correlation among the roles we use the correlation table for that it also we use the correlation and type of the correlation of the role of the system by this we define these role of the system. If our task group matches high nodes we called it strongly positive and show it as A^{++}. If our task group matches, medium nodes we called it partial positive and show it as A^{+}. If our task groups no matches, medium nodes we called it partial negative and show it as A^{-}. If our task groups no match's high nodes we called it sufficiently positive and show it as A^{--}.if we can do the matches on to our criteria we put in the blank space. We use the no correlation if put blank in the table of the system.

3.6 Fabricated Model:

It takes the QSMSR Architectural Configuration as an input and allocates the architectural model .it compares the architectural sub configuration with architectural sub model and then the model selected which is best in the best. It define the map between requirements and design of the system by correlation we get the system. This is our final process of the model we check the correlation and which is strongly connected to the system is selected as architectural design of the system. Than we develop the system architectural design according to that.

Results:

In below figure we have input SR model and shoe the results to refine the figures of our goal we refine our goal and the refine goals are our out put.

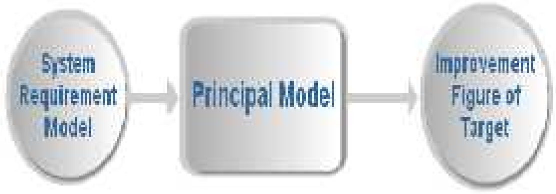

Figure 14: SR Model to Refinement of goal

In this figure the refinement of goal process by the recursively and used as a input of the system and the results are the roles of the system.

Figure 15: Refinement of goal to roles

In below figure the roles used as the input and the resultant is the role iteration graph.

Figure 16: Roles and SR model to Roles interaction graph

In below figure the role integration graph and architecture style is used as the input and the resultant role interaction matrix of architectural style.

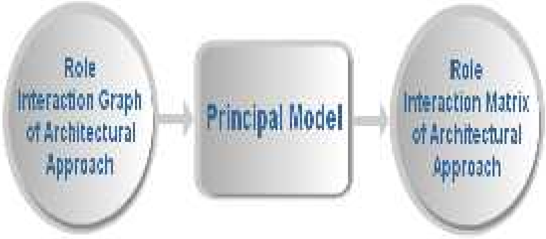

Figure 17: Architectural style and its role interaction to role matrix

Figure 18: Qualitative Classified model as centrality output

In above figure showed the working of the qualitative classified model that analysis the central actor of the system by using the degree and closeness of centrality table.

Figure 19: Qualitative Classified model as similarity correlation

In above figure showed the working of the qualitative classified model that analysis cluster of the match that role with different architecture styles and define which roles of the system are the similar to each other.

Figure 20: Qualitative Classified model as correlation analysis

In fabricated model Architectural Configuration as an input and allocate the architectural model .it compares the architectural sub configuration with architectural sub model and then the model selected which is best in the best.

Discussion:

To eliminate the gap between requirement and the system design we use the QSMSR framework. QSMSR first of all describe the system than divide the system into sub modules and describe the each sub model and match that module with architecture styles. Than it purpose the architecture design of the system. by doing this QSMSR divide its work further into two model principal model and qualitative model each sub model has its own we discus the each phase of QSMSR model followed by the figures.

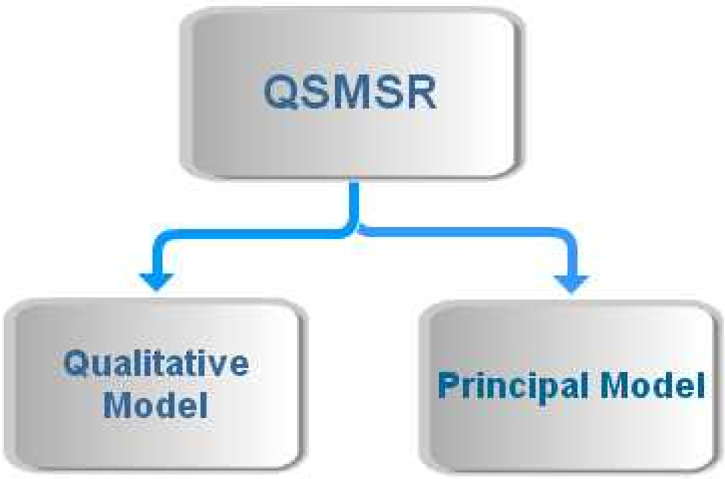

Figure 21: QSMSR basic models

In above figure the main structure of the QSMSR model is discus that the QSMSR divide into two sub models that are principal model and the qualitative model.

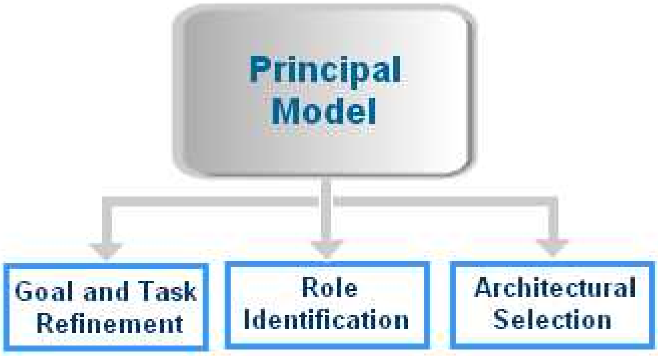

Figure 22: Principal Model

The figure above provides the detail about the principal model sub activities. First activity of the principal model is goal and task refinement. Than role identification and the third and last activity perform by principal model is architectural selection.

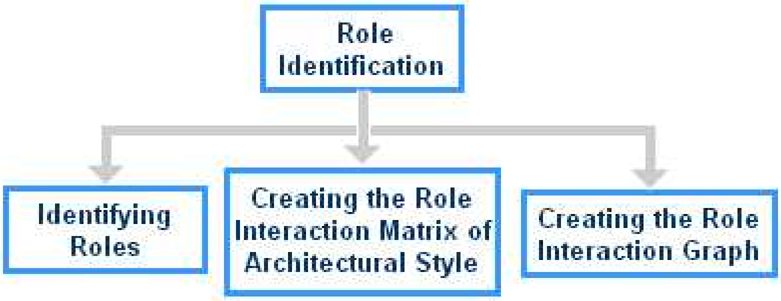

Figure 23: Role identification

The role identification is further sub activity as given above figure. Firstly identify roles and than creating the role interaction matrix of architectural style.

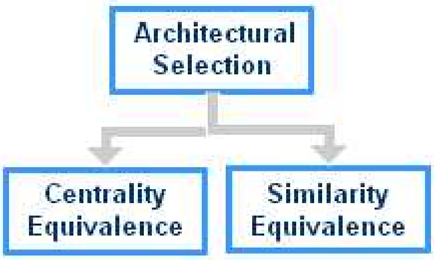

Figure 24: Architectural selection

The architectural selection is the last activity of principal model in this there further two sub activities centurial equivalence and similarity equivalence. It proudest the role interaction graph.

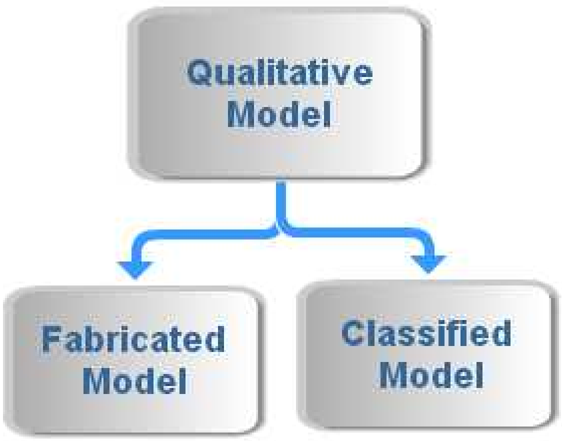

Figure 25: Qualitative model

The qualitative model is also further divided in two sub model are called fabricated model and classified model.

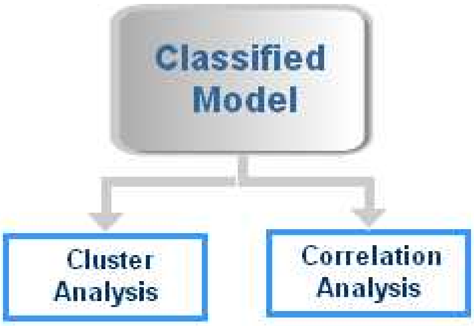

Figure 26: Classified models

The classified divide its activities further in two sub activities correlation analysis and cluster analysis.

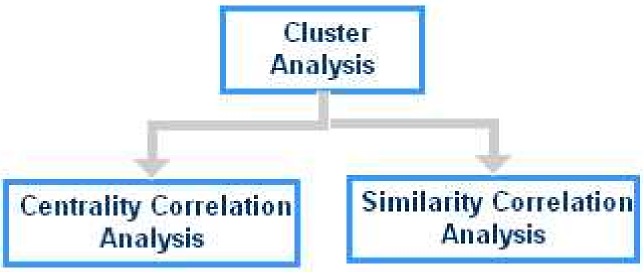

Figure 27: Cluster Analysis

The cluster analysis activities are Similarity Correlation Analysis, Centrality Correlation Analysis in this don the correlation and cluster analysis with help of formulas.

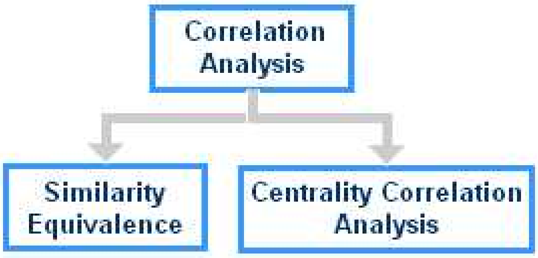

Figure 28: Correlation Analysis

The correlation analysis activities are Similarity equivalence, Centrality Correlation Analysis in this don the correlation and cluster analysis with help of formulas.

References:

Chung, L., Nixon, B. A., Yu, E. and Mylopoulos, J. (2000): “Non-Functional Requirements in Software Engineering”. Kluwer Publishing.

Rodolfo Carneiro Cavalcante, Ig Ibert Bittencourt, Alan Pedro da Silva, Marlos Silva, Evandro Costa ,Robério Santos 10 October 2011.

Alencar, F.M.R., Castro, J.F.B., Cysneiro, G.A.A., Mylopoulos, J. (2001): “Integrating Organizational Requirements and Object Oriented Modeling”. 5th IEEE International Symposium on Requirements Engineering, Toronto.

Allen, J. H.; Barnum, S.; Ellison, R. J.; McGraw, G.; & Mead, N. R. *Software Security Engineering: A Guide for Project Managers*. Boston, MA: Addison-Wesley Professional, 2008 (ISBN: 978-0321509178).

Alencar, F.M.R., Castro, J.F.B., Cysneiro, G.A.A., Mylopoulos, J. (2001): “Integrating Organizational Requirements and Object Oriented Modeling”. 5th IEEE International Symposium on Requirements Engineering, Toronto.

Bastos, L.R.D. and Castro, J.F.B. (2004): “Systematic Integration between Requirements and Architecture”. Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems SELMAS'04. Edinburgh, Scotland, May. LNCS 3390: “Software Engineering for Multi-Agent Systems III”, pp. 85-103.R.Choren et all. (Eds.), Springer-Verlag, 2005.

Christopher Zhong, Scott A. DeLoach 2011.

Manuel, K. and John, M. (2007) Software Architectures as Organizational Structures Department of Computer Science, University of Toronto, Toronto M5S3G4, Canada {mkolp, jm}@cs.toronto.edu

Mead, N. R. "Experiences in ElicitingSecurityRequirements." *CrossTalk 19,* 12 (December 2006): 14-19.

Mead, N. R. &Stehney, T. "Security Quality Requirements Engineering (SQUARE) Methodology." *Software Engineering for Secure Systems (SESS05), ICSE 2005 International Workshop on Requirements for High Assurance Systems*, St. Louis, MO, May 15-16, 2005. http://homes.dico.unimi.it/%7Emonga/sess05.html.

MICHAEL WOOLDRIDGE, NICHOLAS R. JENNINGS, DAVID KINNY Autonomous Agents and Multi-Agent Systems, 3, 285.312, 2000

Scott A. DeLoach& Matthew Miller. A Goal Model for Adaptive Complex Systems. International Journal of Computational Intelligence: Theory and Practice. Volume 5, no. 2, 2010.

Wooldridge, M., Jennings, N.R. and Kinny D. (2000): "The Gaia Methodology for Agent-Oriented Analysis and Design". Autonomous Agents and Multi-Agent System, 3, 285312. Kluwer Academic Publishers, Netherlands.

Yu, E. (2001): "Agent Orientation as a Modeling Paradigm". Wirtschaftsinformatik. 43(2), April- pp.123-132.

Zambonelli, F., Jennings, N., Wooldridge, M. (2001): "Organisational Abstractions for the Analysis and Design of Multi-Agent Systems". In: Ciancarini P., Wooldridge, M. (eds.): Agent-Oriented Software Engineering, LNCS 1957, Springer- Verlag, pp. 235 – 251.

Zambonelli, F., Jennings, N.R., and Wooldridge, M. (2003): "Developing Multi-agent Systems: The Gaia Methodology". ACM Transactions on Software Engineering and Methodology, vol. 12, no. 3, pp.417 --470, 2003.